Towards a Methodology for Profiling Cyber Criminals

Leonard Kwan, Pradeep Ray and Greg Stephens
School of Information Systems, Technology and Management
University of New South Wales, Sydney, Australia

Abstract

The progress of future e-business and e-commerce will depend on the ability of our legal institutions to protect general users from cyber crimes. While there has been substantial progress in the development and implementation of tools for detecting and preventing cyber attacks, there is a lack of effective methodologies to prosecute the perpetrators of cyber crimes (cyber criminals). Consequently, many cyber crimes go unpunished and many intrusion detections tools go unutilized. Hence there is a need for holistic methodologies that can help organizations collect legally valid evidences from cyber crimes so that appropriate actions can be taken against cyber criminals. This paper presents an approach for this objective by using honeynets.

1. Introduction

Cyber security is emerging as one of the major factors affecting the growth of e-business and e-commerce. The solution to cyber crime has a number of aspects, namely detection, prevention and law enforcement. Current information security mechanisms predominantly focus on detection and prevention. Currently cyber laws and their enforcement mechanisms are not clearly defined in most countries. Consequently, many cyber crimes and cyber criminals can not be prosecuted effectively. The prosecution of the intruders, however, is the only long term solution to stemming the tide of increasing cyber crimes. To assist in this process, a new branch of forensics has evolved – namely digital forensics.

In this paper, we propose a digital forensics methodology based on data collection mechanisms of honeypots for the purpose of profiling cyber crimes and cyber criminals for cyber crime-cyber criminal correlation. That may lead to more effective action against cyber criminals.

Although there have been substantial advances in Internet security tools and techniques (e.g., firewalls, anti virus, Intrusion Detection Systems etc.), they may not be adequate for gathering evidence to prosecute cyber criminals. Honeynets are a supplementary resource that can be included in the security architecture of a network; they are dedicated networks comprised of honeypots, systems that are designed to be probed, attacked and compromised. Their primary purpose is to collect information on the activities of cyber criminals [1]. This is made possible by purposely constructing a vulnerable and insecure network so that one can attract attackers for subsequent study. It is, however, important to note that a honeynet does not replace existing security measures (e.g., firewalls, IDS etc.) – rather they serve to enhance them by supplementing additional network intrusion data.

This paper based on the thesis [13] is organized as follows. The next Section briefly introduces the field of digital forensics. This is followed by a discussion of criminal profiling as used today by law enforcement agencies. We then discuss cyber crime profiling in four dimensions, namely breadth, depth, vulnerabilities, data collection tools (e.g., honeypots) and cybercrime-cybercriminal correlation.

2. Digital Forensics

Digital forensics is defined as "The use of scientifically derived and proven methods towards the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations"[4]. Due to the relative infancy of digital forensics, there exist a number of preliminary frameworks that are evolving to address the digital investigations process.

The Digital Investigation Process framework represents an effort by Beebe & Clark (2004) to

1530-1605/08 $25.00 © 2008 IEEE

"parsimoniously encapsulate all phases and activities outlined in prevailing models presented to date" [2]. Whilst single tiered frameworks suited the preliminary development and infancy of digital forensics, the "complexity level associated with the digital investigative process necessitates a multi-tier, hierarchical framework". The framework includes six stages, namely [2]:

- Preparation
- Incident Response
- Data Collection
- Data Analysis
- Presentation of Findings
- Incident Closure

Preparation also referred to as readiness in the Carrier & Spafford (2003) framework, helps to maximize the availability and quality of digital evidence - an imperative component of digital investigations [3]. Unfortunately, digital evidence is frequently "damaged or destroyed during standard containment, eradication and recovery activities" [2]. Preparation or forensic readiness has been suggested to cover two objectives – maximizing the ability to collect digital evidence while minimizing the cost of incident response [8]. Activities in the preparation phase may include personnel training, the development of evidence preservation and handling procedures, the development of technical capabilities as well as information retention plans.

Incident response is described by Beebe and Clark (2004) as "the detection and initial, pre-investigation response to a suspected security incident". This phase aims to establish a driving force for the forensic investigation that will steer the later stages of the forensics process. Key activities that may be conducted include the identification of suspicious or unauthorized activity, the validation of a security incident, the formulation of an investigation and response plan as well as an assessment of damages.

Beebe and Clark (2004) explain that this phase entails the collection of "digital evidence in support of the response strategy and investigative plan" as set out in the incident response phase. While some data may already have been collected in the incident response phase to validate and determine the impact of the incident, the formal data collection phase occurs following the decision to pursue a digital investigation "regardless of its scope or anticipated legal or administrative actions". Data collection activities may include the acquisition of removable media, host and network based evidence, the certification of the integrity and authenticity of the digital evidence as well as the secure packaging, transport and storage of the digital evidence.

Data analysis is possibly the most complex and time consuming phase in the digital forensic investigation process. Beebe and Clark (2004) suggest that confirmatory analysis and/or event reconstruction to be the main components of the data analysis phase. Confirmatory analysis serves to "confirm or refute allegations of suspicious activity" while event reconstruction helps to answer the "who, what, where, when, why and how" type questions. Several iterations of examination and analysis of the data collected in the data collection phase may be required to support a theory. Activities might include transformation and consolidation of data to facilitate easy of management and analysis, data extraction, executable analysis and even decryption of data.

The presentation of findings is for the purpose of communicating the relevant findings to a variety of audiences which can include, but are not limited to management, legal personnel and law enforcement. Several researchers ([6], [2]) have emphasized that the presentation should be written and presented in layperson's terms using abstracted terminology which references the specific details. More often than not, technical reports produced by digital forensic analysts while capturing and documenting the findings are not of high value as they cannot be easily understood by those who will act upon the findings.

The purpose of incident closure is four fold and entails the critical review of the forensic investigation, the formulation and actioning of decisions that arise from the presentation of findings, the disposal of the evidence (returning to owner, destroying it, or cleaning it for re-use) and finally the preservation of all information relating to the incident.

To these first tier steps, Beebe and Clark (2004) add that it is necessary for additional sub-phases to "provide adequate detail needed in the overall framework". They reinforce the notion that the digital investigative process is complex enough to necessitate a "multi-tier, hierarchical framework with objectives-based sub-phases that are applicable to various layers of abstraction [2]. The objectives-based sub-phases (OBSP) are integrated with the first tier framework as shown in Figure 1.

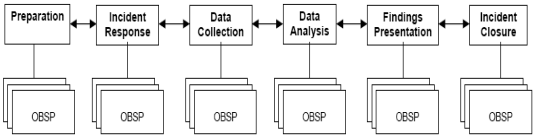

Figure 1 – Two-Tier Digital Investigations Process Framework [2]

The Beebe and Clark (2004) framework is a practical approach to digital forensics investigations and the aforementioned steps best match those taken in actual investigations. Furthermore, with a basis in existing physical crime investigation theory and a technology neutral standpoint, the digital investigations process framework offers unique benefits over the previously proposed frameworks.

In our research study we focus predominantly on the data analysis phase of the aforementioned framework. We will begin with a quick discussion on the notion of criminal profiling, the basis of our data analysis.

3. Criminal Profiling

.

The first problem encountered in this area is that of the answer to the question "what is cybercrime?". The term has become common but there seems not to be a commonly accepted definition. The broad definitions offer little insight. This is exacerbated by the fact that the term itself is a social one rather than one defined in law. Garlik Cybercrime Report, definitions of cybercrime include [14]:

- the use of any computer network for crime
- any criminal offence committed against or with the help of a computer network

The concept of criminal profiling is quite common in the criminal justice system and its roots can be traced back to the end of the 19th century [7]. Law enforcement have embraced profiling as an investigative aid since its humble beginnings and it continues to be highly regarded today [12].

While the nature of evidence is evolving from physical to electronic, Rogers (2003) explains that criminal profiling can still play a vital role in the digital forensics process and that "we need not totally abandon traditional investigative approaches, but merely allow them to evolve" [7].

There are two predominant profiling methods at our disposal – inductive and deductive profiling. In our research study, we employed a deductive profiling approach which is arguably more evidence based than the inductive approach. We will be aiming to address the trial phase of Turvey's (1999) Behavioral Evidence Analysis (BEA) profiling technique. Essentially, the trial phase aims to associate a known crime with a known offender [12].

To do this, however, we must first be able to profile the offender and to a certain extent, the crime. This is facilitated by the investigative phase of Turvey's (1999) BEA technique. Here, we have a known crime, but an unknown offender. As such, it is necessary to formulate profiles for both the offender and the crime they perpetrated.

A number of previous studies have explored the issues of perpetrator profiling using a variety of methods. Kjaerland (2005) used data from actual attacks to develop taxonomies based on categories and the relationships between them [15]. Others have used a variety of methods using alert regression profiles [16].

In our research, we have developed several key concepts that assist in the process of cybercrime profiling, cybercriminal profiling and subsequently cybercrime-cybercriminal correlation. The cybercrime profiling and cybercriminal profiling address the trial aspect of the BEA model while the cybercrime-cybercriminal correlation addresses the trial phase. These will be further discussed below.

4. Cybercrime Profiling

With a considerable amount of collected data at our disposal, we proceeded to construct mechanisms for assessing this data. As proposed by McGrew and Vaughn (2006) an attack profile could contain useful information about motivation, breadth, depth, sophistication, concealment, attacker(s), vulnerabilities and tools [5]. To demonstrate the benefits of profiling, we focused on four elements in our research study. It was decided that, in the absence of established and tested metrics, to use breadth, depth, vulnerabilities and tools. The chosen elements were selected as a suitable starting point for the assessment of the data with a view to refining the metric, if required, in future developments of the algorithm. We are not suggesting that this is a complete set for profiling but rather a good starting point.

4.1 Breadth

Breadth is a metric used to measure the scope or range of infiltration of the attack. In our research study, we have established two key abstractions of depth – host breadth and network breadth. The host breadth measures the scope of infiltration on a singular host whilst the network breadth measures the scope of infiltration on the network.

Host and network breadth can be measured through network connection analysis. Host breadth can be determined by analyzing connections from the perpetrator to the various ports on the target system. In contrast to host breadth, network breadth can be determined by analyzing connections from the perpetrator to various hosts on the target network.

For example, a port scan of a host, to determine open ports/services on the target machine, would gain a high rating for host breadth; however, due to the lack of activity directed against other hosts on the network, it would be classified a low network breadth attack. Conversely, a host scan, to determine online machines on the target network, would score a high network breadth rating, yet a low host breadth rating. Furthermore, a systematic and exhaustive scan of all open ports and services on a network would yield both a high host and network breadth rating.

The Generation III Honeywalls provide the functionality to [1]:

- Extract network packet captures from the file system;
- Extract simplified network connection history from the Hflow 'argus' table.

4.2 Depth

Depth is used in a similar way to breadth, however, its focus is to measure the extent or degree of infiltration of the network intrusion. There are two abstractions of depth – host depth and service depth. Host depth represents the level to which an infiltration penetrates the host; thus providing a general determination of the host penetration. Service depth, on the other hand, measures the degree of infiltration of a particular service; this is capable of revealing more detailed information concerning the exact activities conducted by the attacker.

To illustrate depth, a protracted connection to a host could indicate that an individual has successfully compromised the host or is, in fact, snooping around the system in preparation for a future attack. A prolonged connection from an attacker could indicate a high degree of infiltration as it appears that the attacker and target hosts are actively communicating to each other. To yield a measure of service depth, we can analyze the logs obtained from various services like database, FTP, SSH and others. By looking for key accounting records like user authenticated, file access and other chronicled activity, it is possible to determine the depth to which an attacker has infiltrated a particular service.

It is accepted that using host depth may not be the ultimate choice; but rather an expedient one that will require future refinement. Eventually this kind of work will lead to provision of evidence for legal proceedings and consequently more work is required to establish suitability of that evidence. Other factors such as network traffic conditions and device speeds will affect connection times. In the future more complex algorithms may use network topology or routing traces if these are deemed more valid with respect to evidence provision.

One way of determining host depth is by analyzing the connection times from the perpetrator to the target system. The determination of service depth, however, is significantly more difficult. The analysis of service depth is reliant on the verbosity of the logging facilities of the service. Current day logging facilities are varied in their ability to provide suitable and thorough accounting data required to facilitate cybercrime profiling.

To assess service depth in our honeynet, we can consult service logs residing on individual honeypots. To establish a high level assessment of host depth, we can analyze the network packet captures and connection history located on the Honeywall [1].

4.3 Vulnerabilities

Profiling of network incidents has historically entailed the analysis of the vulnerabilities exploited. In our cybercrime profiling concept, vulnerabilities remains a key component of the overall field view.

Vulnerabilities are the flaws and weaknesses in a system that allows an attacker to compromise it. As an independent measure, it does not contribute extensively to the cybercrime profile. However, when used in conjunction with elements such as sophistication and attackers, the amalgamation can be used to increase the precision of the cybercrime profile.

The exploitation of vulnerabilities can often be detected through the use of intrusion detection

systems. These systems are chiefly signature-based systems and are generally incapable of identifying new exploits. The exploitation of common vulnerabilities can enhance our determination of an attacker's modus operandi, however, does not specifically indicate the level of sophistication of an attacker. In the case where an attacker exploits an "unseen" or "new" vulnerability, it can be assumed that the attacker possess a high level of sophistication.

In our honeynet architecture, we can assess vulnerabilities by analysing the packet captures collected by the Honeywall [1].

4.4 Tools

The final component of our cybercrime profile is tools. Often tools are simply perceived as the software that is utilised to exploit vulnerabilities. Our conception of tools actually encompasses all the software and all the hardware that is used in an attack. On the software side, this includes root kits, backdoors, attack scripts and even normally innocuous software like ftp, ssh, ping and wget. Hardware includes, but is not limited to the attacker's system as well as their internet connection.

By analyzing the tools used, we can formulate a more accurate assessment of the attack's motivation and sophistication. For instance, the extensive use of file transfer software like ftp or database query clients like sqlplus or mysql can be linked to instances of corporate espionage. The reasoning is that database query client usage may indicate that intruders are attempting to export/query confidential data (e.g. financial and personal data).

To assess the tools that were used, we can consult several tables within the Honeywall Hflow schema. The Sebek related tables; 'sys_read', 'sys_open' and 'sys_socket'; can reveal the commands executed on our honeypots. Passive fingerprinting tools like p0f store the data collected about source systems in the 'os'. Among the data captured by p0f are the operating system of the attacker, the system architecture and even their network connection [1].

4.5 Cybercrime-Cybercriminal Correlation

In order to broaden digital forensics we employ two approaches to the handling of cybercrime profiles. Having described the contents of our proposed cybercrime profiles, they can be associated with a known cybercriminal or left standalone for later cybercrime-cybercriminal correlation.

We propose to use a conceptualization of a cybercriminal as a container for holding cyber crimes. This is quite a legitimate abstraction as the actions that an individual executes can actually reveal characteristics of the individual. By associating cyber crimes with their perpetrator, we can actually add precision and detail to the cybercriminal's profile.

To perform cybercrime-cybercriminal correlation it is necessary for us to perform a side-by-side comparison. Recalling that cybercriminal profiles are comprised of cybercrime profiles, we can compare a "test" cybercrime profile with each of the cybercrime profiles in the cybercriminal profile. To assist in this comparison, we have devised a similarity index formula as follows.

$$\text{Similarity Index} = \alpha\left(1 - \frac{|S_i - S_j|}{S_i + S_j}\right)$$

This formula calculates an index between 0 and 1 for a scored element. S_i represents the "score" of the element from profile 'i' and α is a number between 0 and 1 that can be applied as a "weighting" to modify the calculated similarity index. Essentially, the similarity index will tend towards 0 the greater the difference in the score of the element and vice versa. An element with a similar score in both profiles will score an "unweighted" similarity index close to 1. Conversely, dissimilar element scores will score an "unweighted" similarity index close to 0.

As the similarity index only calculates the similarity between comparable elements, we are unable to directly use this for correlating a cybercrime to a cybercriminal. This correlation can be performed on two levels – a high-level profile-level abstraction or a low-level element-level abstraction. This will be further discussed in the cybercrime-cybercriminal correlation tool in the implementation section.

5. Implementation

The Honeynet Project, one of the primary driving forces behind many honeynets today, is a research organisation "dedicated to learning the blackhat community's tools, tactics and motvies and then sharing any lessons learnt" [11]. Since its inception, they have produced several products, most recently the 'Roo' Honeywall Gateway which is a key component of the

Generation III honeynet. By providing the key facilities of data capture, data control and automated alerting, the Generation III honeynet is an extremely capable mechanism for capturing network intrusion data.

In our research, we utilised a Generation III honeynet implementation and hosted it on an educational network. Over the course of two months, we collected a sizable quantity of data. Over 1.2 million distinct connections were observed from over 200,000 unique hosts. In excess of 200,000 intrusions were detected by Snort during this time frame and the statistics are staggering – each unique host perpetrated approximately one intrusion and one in six connections contained an intrusion [10].

A prototype system was created to demonstrate how it was possible to:

- Create cybercrime profiles based on the aforementioned profiling elements;
- Create cybercriminal profiles based on the profiled cyber crimes; and
- Perform cybercrime-cybercriminal correlation.

As a proof-of-concept, our prototype sought only to profile cyber crimes based on breadth, depth, tools and vulnerabilities.

5.1 Cybercrime Profiling Tool

The objective of the cybercrime profiling tool is simply to define from a high level; its primary purpose is to gather all pertinent information relating to the cybercrime profile and then perform, if necessary, an initial analysis of this data.

For the purposes of our prototype, we chose to analyze four key elements – breadth, depth, tools and vulnerabilities. Roughly categorizing our elements based on their source data, both abstractions of breadth and the host depth would fall under connection data, tools and service depth under Sebek data while vulnerability data would remain standalone.

As assessing vulnerabilities and performing connection analyses is a time consuming process, they will be executed on cybercrime profile creation. Generally, it is unlikely that the assessment will change over time, hence the results can be stored persistently and a re-analysis can be conducted as required. On some occasions, signatures for new vulnerabilities may be released and profiles may need to be re-analyzed to search for these vulnerabilities.

While the benefits of a Honeywall data collection apparatus are clear, the stored data is regrettably distributed over several tables in the Hflow schema and inside the file system. To create our cybercrime profile, we can extract the connection data and packet captures and populate them into our cybercrime profile. Following this, we can execute a vulnerability assessment and perform connection analyses to search for activity like port scans, network scans and others [1].

To extract the processed connection data, we queried the 'argus' table within the 'walleye_0_3' schema within Hflow. The raw packet captures were extracted using the 'pcap_api' script.

To determine the vulnerabilities exploited, we used Snort to analyze the packet captures. Snort logged detected exploits to the "alert.ids" file and we then used regular expressions and string processing to extract data like the signature id, generator, revision and name.

As we were using Windows 2000 honeypots within the honeynet, we were unable to capture any data in the 'sys_open' table as the Windows port of Sebek did not support it. We were, however, able to relate the sys_read() activity back to the connections stored in the 'argus' table by performing several joins through the tables of 'process' and 'sys_socket'.

This resulting data was loaded into our profiling database for persistent storage. The cybercrime profile can then be used by our cybercriminal profiling tool and cybercrime-cybercriminal correlation tool.

5.2 Cybercriminal Profiling Tool

Recalling that we proposed that cybercriminal profiles were comprised of cybercrime profiles, we can conceptualise the cybercriminal profile as a container holding cybercrime profiles. This structure is simple to create in a relational database. Details are available in [13].

6. Cybercrime-Cybercriminal Correlation Tool

It defines the rate of increase in the linear weighting and 'A' and 'k' combined defines the rate of increase in the exponential weighting.

These parameters are stored persistently in our profiling database along with an event signature. This event signature can simply be a search string, a complex regular expression or even a command name. We can search for the number of occurrences of the event signature in

the Sebek data or service logs and then calculate the cumulative weight for a particular event.

These event signatures and weighting parameters are also associated with a grouping entity; usually event signatures are grouped by their generator, for instance, FTP events reside in a FTP group, while bash shell console history could reside under bash_history. These groups are stored in table within our profiling database and their primary use is for aggregating the cumulatively weight of the group's events. The weight of the group is calculated by summing the cumulative weight of each of the group's events. This weight, also referred to as the group score, can then be used a quantitative assessment of the group.

In addition to Sebek event weighting, we can also perform some connection analysis to assist in our evaluation of breadth and host depth. There are several superficial calculations that we can use to determine and compare host breadth, network breadth and host depth as well as some more complicated assessments like detecting host scans and port scans.

SQL statements can be executed to perform and assist in various connection analyses. Examples of SQL queries that can be executed to assist in connection analyses are shown below.

Analysis	SQL to Execute
Unique Ports Probed	SELECT count(*) from "connection_extract" GROUP BY "dst_port";
Unique IPs Probed	SELECT count(*) from "connection_extract" GROUP BY "dst_ip";
Total Connection Time from Attacker to Target	SELECT "src_ip", "dst_ip", SUM(end_sec - start_sec) as "conn_time", count(*) as "conn_count" from "connection_extract" GROUP BY "src_ip", "dst_ip";
Port Scans	SELECT "src_ip", "dst_ip", "dst_port" from "connection_extract" ORDER BY "src_ip", "dst_ip", "dst_port";
Host Scans	SELECT "src_ip", "dst_ip" from "connection_extract" ORDER BY "src_ip", "dst_ip";

Using these SQL statements, we can determine superficial assessments of breadth by counting the number of distinct ports (services) and IP addresses (hosts) probed. Furthermore, we can search for sequences of IP addresses and ports, commonly known as port scans and hosts scans. This search process can be customised to detect scans of common ports, that is ports of commonly used services like http (80), ftp (21) etc.

To compare the aforementioned scans, we can use an adapted similarity index calculation to compare the lengths of the sequences. To determine the lengths of the sequences, we can look for continuous blocks of ports or IP addresses. When a break is observed, the current sequence ends and new sequence begins.

We have also built some resilience into this sequence analysis. Of particular note is that the first and last addresses in a network address block are the network address and broadcast address respectively. These addresses, like 10.0.0.0 and 10.0.0.255, will never appear in a network scan. However, a network scan could span over several network address blocks. For example, if an attacker were to scan the networks 10.0.1.x and 10.0.2.x, the sequence near the transition of the network blocks would appear like the following: 10.0.1.253, 10.0.1.254, 10.0.2.1, 10.0.2.2.

In addition to this, it is possible that some corrupted packets are dropped which could result in breaks in the sequence. As such, we have implemented two customizable thresholds. The first defines the minimum number of elements a sequence must contain before it is deemed significant and worthy of inclusion. The second defines the size of the acceptable gap between the two elements in a sequence.

Armed with the lengths of the port scans and host scans, we can now substitute these values into our adapted similarity index formula shown below. To use this formula, we substitute $N_{s,i}$ with the length of each sequence of type 's' from profile 'i'.

$$\text{Similarity Index}_{\text{Sequences}} = \frac{\sum_{\text{all sequences in 's'}} \left(1 - \frac{|N_{s,i} - N_{s,j}|}{N_{s,i} + N_{s,j}}\right)}{\text{total number of sequences}}$$

If the two profiles do not contain the same amount of sequences for each sequence type, we can use "fillers" – a sequence of the same type but with a sequence length of 0. Effectively, this will mean that there is big discrepancy and the similarity will be calculated as 0.

Unfortunately, the similarity index calculation above may not compute the largest similarity index which is what we are looking for. For example, by comparing the sequence lengths in different order, a different similarity index could be computed. The formula above treats the sequence lengths as a list – order is important. To ensure that we compute the largest similarity index, it is necessary for us to treat the sequence lengths as a set – where order is not important. To do this, we can permutate all possible orders of the sequence lengths for one of the profiles.

Unfortunately, this permutation process is extremely time and resource intensive and takes a considerable amount of time to process. If there are more than 13 sequences of a particular type, profiling time is unacceptable as there are over 6.2 billion trials to test before the largest similarity index can be determined. As we are only prototyping our implementation, it was decided that if there were over 11 instances of a particular sequence type, the list would simply be sorted. Regrettably, it means that we may not necessarily have calculated the highest similarity index.

This process must be repeated for each sequence type and these can include inbound port sequences, outbound port sequences, inbound IP sequences, outbound IP sequences, inbound common ports and outbound common ports.

Finally, to compare the vulnerabilities exploited we can compare the results of the Snort analysis. During the cybercrime profiling stage, we parsed the packet captures through Snort and then analysed the "alert.ids" file produced.

To calculate the similarity index for vulnerabilities, we first extract the list of the vulnerabilities exploited from each profile. The intersection of these two lists was taken to determine how many vulnerabilities were common and this was subsequently divided by the number of vulnerabilities in the union of these two lists.

The resulting number between 0 and 1 signifies the similarity index of the vulnerabilities exploited. The formula for this calculation is shown below. Vulnerabilities_i signifies the list of vulnerabilities exploited in profile 'i'.

$$\text{Similarity Index}_{\text{Vulnerabilities}} = \frac{Count(\text{Vulnerabilities}_i \cap \text{Vulnerabilities}_j)}{Count(\text{Vulnerabilities}_i \cup \text{Vulnerabilities}_j)}$$

More details are available in [13].

7. Conclusion and Future Work

In this paper we presented a methodology for correlating cyber crimes with known cyber criminals. Our methodology adopts several of McGrew and Vaughn's (2006) attack profiling elements for the purposes of network intrusion profiling [5]. The resulting cybercrime profiles could then be associated with a known cybercriminal, ultimately facilitating cybercrime-cybercriminal correlation.

To assist in the process of correlation, we developed several original mechanisms including a weighting mechanism for assessing Sebek and log data. In addition to this we devised several similarity index formulae that assist in the correlation process.

As a proof-of-concept, it was sufficient to demonstrate some of the elements of McGrew and Vaughn's (2006) attack profiling elements. Unfortunately, for us to be able to profile elements like motivation and sophistication, it is necessary for us to have a large knowledgebase or expert system to assist in the assessment.

Sophistication which is relative requires a benchmark for successful evaluation. Without historical data regarding the sophistication of attacks, it is difficult for us to judge the relative sophistication of a new attack. Thus it is also difficult to compare the sophistication of attacks.

While elements like motivation may remain qualitative, it could be possible to group the motivations of attackers into several broad categories. For example, Turvey (1999) developed "general motivating topologies for Internet-related criminal activity"; he proposed five general topologies including power assurance, anger retaliation, sadistic, opportunistic and profit. Rogers (2003), also grouped offenders into several categories and suggested that these could include greed, revenge, anger, perversion, politics and a desire for power. While these are somewhat broad categories, most root causes of an intrusion – that is the motivation – will fit inside one or more of these categories.

While it is possible that the motivation of an attack can fit within these categories and compared to some extent, the difficulty lies in automating the assessment. Without a knowledgebase or expert system which can correlate the motivation with an attack signature, this will remain to be a hurdle.

Finally, due to resource constraints we were unable to implement a unique method of assessing an attacker's sophistication.

However, this work has illustrated a holistic approach to tackle cyber crimes from the perspective of law enforcement. More work is required to validate such a methodology in a commercial environment. Its success will pave the way for law enforcement agencies to tackle cyber crimes and cyber criminals within the existing legal environment. More details are available in [13].

Acknowledgement:

The Research reported in this paper was partially supported by the Australian Research Council (ARC) Discovery Grant DP0451650.

References

[1] Balas, E., Honeynet Data Analysis: A technique for correlating sebek and network data, http://www.dfrws.org/bios/day2/Balas_Honeynets_for_ DF.pdf, last accessed November 2005

[2] Beebe, N.L. & Clark, J.G. (2004) 'A hierarchical, objectives-based framework for the Digital Investigations Process', Digital Forensics Research Workshop (DFRWS), Baltimore, 2004

[3] Carrier, B. & Spafford, E. (2003) 'Getting Physical with the Digital Investigation Process', International Journal of Digital Evidence, Volume 2, Issue 2

[4] DFRWS (2001) – Digital Forensic Research Workshop (2001) A road map for digital forensic research: Report from the First Digital Forensic Research Workshop (DFRWS), New York

[5] McGrew, R. & Vaughn, R.B. (2006) 'Experiences With Honeypot Systems: Development, Deployment, and Analysis', Hawaii International Conference on System Sciences, 2006

[6] Reith, M. & Carr, C. & Gunsch, G. (2002) 'An Examination of Digital Forensic Models', International Journal of Digital Evidence, Fall 2002, Volume 1, Issue 3

[7] Rogers, M. (2003) 'The role of criminal profiling in the computer forensics process', Computers and Security, May 2003, Volume 22, Issue 4

[8] Rowlingson, Robert "A Ten Step Process for Forensic Readiness," International Journal of Digital Evidence (2:3), Winter 2004, pp 1-28.

[9] Shinder, D.L., 2002. Scene of the cybercrime: Computer forensics handbook. Rockland MA: Syngress

[10] Snort, 2006, SNORT- The de facto standard on Intrusion Detection and Prevention, www.snort.org

[11] Spitzner, L. (2003) 'The Honeynet Project: Trapping the Hackers', IEEE Security and Privacy, March 2003, Volume 1, Issue 2

[12] Turvey, B (1999), Criminal profiling: An introduction to behavioural evidence analysis, Academic Press, London

[13]Kwan, L. (2006), Fingerprinting Computer Criminals through Computer Intrusion Profiling and Correlation, BIT Honours Thesis, School of Information Systems, Technology and Managenment, University of New South Wales, Australia, 2006

[14] Garlik, UK, Cybercrime Report, . https://www.garlik.com/press/Garlik%20Cybercrime%20Report.pdf, Accessed 11 September, 2007.

[15] Kjaerland, Maria. (2005). A Classification of Computer Security Incidents Based on Reported Attack Data. Journal of Investigative Psychology and Offender Profiling Vol **2**: 105–120.

[16] Phoung Yarng, Pradeep Ray and Danny Maher, "Profiling Cyber Attacks using Alert Regression Profiles", Proceedings of the IEEE Global Telecommunications Conference (Globecom2003), San Francisco, USA, December 2003